	Page
27/11/24	
	LAB - 10
<u>a)</u>	Danie to the internal Commission and
(4)	Demonstrate inter process Communication and deadlack.
	soudateick.
	class Q {
	int n;
	boalean nalue Set = false;
	synchronized int get () {
	entile (! nalue set)
	try (
	System. out. print la ("In Consumer Waiting \n");
	mait();
	3 Comment to the later of the l
	catch (Interrupted Exception e) {
	System. out. print ln (" 9 nterrupted Exception caught");
	}
	System. rout. println ("Grot:"+n);
	nalue Set = false;
	System. out- print la ("In Intimate Producer \n");
	notify ();
	veeteven n;
	}
	synchronized void put (int n) {
	mile (nalue Set)
	try!
	System. out print In ("Producer Waiting");
	wait ();
	J
	System out println ("Interrupted Exception caught");
	3



```
this n = n;
 nalue Set = true;
System. out. print ln ("Put:"+n);
System - out. print ln ("Intimate Customer");
notify (); }
 class Bracedure implements Runnalele {
 Brocedure
Producer (Q, q) {
  new Threads (this, "Preoducer"). start ();
public recid run () [
 int i = 0;
   ruhile (i < 15) {
  g. put ( ; ++); }
 class Consumer implements Runnalile {
   Consumer (Q, q) {
   new Thread ( this, "Consumer") - stort ();
```

```
Julilia void run () [
int i = 0;
 enhile (i <15) {
   int r = q. get ();
 System - rout - print ln (" consumed "+ r);
class PC Fined {
public static maid main (5 trung args []) {
 Q qy = new Q();
 new Producer (a);
 neue Consumer (q);
 System- rout print In ("Press Contral-C to stop");
```



Put: 1 Crot: 1 Put = 2 Crot: 2 Put = 3 Crost = 3 Put: 4
Crot: 4
Put: 5
Crot: 5



27/11/24	
	Deadloch
	Plaatoch
	class A {
	sepretronized void foo (Bb) {
	String name = Thread. current Thread (). get Name ();
	System. out. print ln (name + "entered A. foo");
	try {
	Thread. sleep (1000);]
	cotch (Enception e) (
	System. out- print ln (" A Interrupted");
	System. out- print In (name + " trying to call B. last ()");
	b. last ();
	}
	Void last () {
	System. west. print ln ("Inside A. last"); }
	3
	class B {
	synchronized void box (A a) {
	String name = Thread- current Thread (). get Name ();
	Septem. rout. print In (name + "entered B. bar");
	try {
	Thread. sleep (1000); }
	cotch (Exception e) {
	Segrtem. sout. print ln ("B 9 nterrupted");
2	

```
System. out. print la (name + " trying to call A. last ");
a. last();
Void last () (
 System. out. print ln ("Inside A. last"); }
class Deadlock implements Runnalle {
  A a = new A ();
  B b = new B();
  Deadlock () {
  Thread. current Thread () - set Name ("Main Thread");
Thread t = new Thread (this, "Racing Thread");
   t-start ();
   a. foo (b);
  System. out print la ("Back in main thread");}
public void sum () {
 b. bar (a);
  Segetem. out- print In ("Back in other thread");
 pullie static void main (String args []) {
new Deadlack (); }
```



Output:
Racing Threads entered B-bar
tours of A Dag
Main Thread entered A. foo
Main Thread trying to call B. last ()
Inside A. last
Back in main thread
2 - The determinents call A last ()
Racing Thread trying to call A. last ()
Inside A. Jost
Back in other thread
~
Seen
41)
and the second s
2x(u)